**IN THE CLAIMS:**

Please amend claims 1, 5, 7, 20, 22, 46-48, 51, 52, 55, 59 and 62 as indicated in the following.

Please cancel claims 3, 4, 6, 9, 10, 18, 19, 21, 24, 53, 54, 63 and 64 without prejudice as indicated in the following.

**Claims Listing:**

1. (Currently Amended) A method comprising:
receiving transformed error correction data;
determining if ~~an error characteristic~~at least one of an underflow error or an overflow error of the transformed error correction data has occurred; and
providing an error indicator when it is determined at least one of an underflow error or an overflow error~~an error characteristic~~ has occurred.

2. (Original) The method as in Claim 1, wherein the transformed error correction data includes multiple channels of transformed error correction data.

3. (Canceled)

4. (Canceled)

5. (Currently Amended) The method as in ~~Claim 4~~Claim 1, wherein extraneous transformed error correction data is ignored when an overflow error has been determined.

6. (Canceled)

7. (Currently Amended) The method as in ~~Claim 6~~Claim 1, further including completing a set of transformed error correction data with predetermined values when an underflow has been detected.

8. (Original) The method as in Claim 7, wherein the predetermined values include zeros.

9. (Canceled)

10. (Canceled)

11. (Original) The method as in Claim 2, wherein providing the error indicator includes generating an interrupt.

12. (Original) The method as in Claim 2, wherein providing the error indicator includes setting a flag.

13. (Original) The method as in Claim 12, wherein separate flags are set for different error identifiers.

14. (Original) The method as in Claim 13, wherein the separate flags are set for different channels.

15. (Original) The method as in Claim 14, wherein the separate flags are polled to determine an error has occurred.

16. (Original) The method as in Claim 15, wherein a driver is used to perform the polling.

17. (Original) The method as in Claim 2, wherein separate error indicators are provided for different error channels.

18. (Canceled)

19. (Canceled)

20. (Currently Amended) The method as in ~~Claim 19~~Claim 1, wherein extraneous transformed error correction data is ignored when an overflow error has been determined.

21. (Canceled)

22. (Currently Amended) The method as in ~~Claim 18~~Claim 1, further including completing a set of transformed error correction data with predetermined values when an underflow error has been determined.

23. (Original) The method as in Claim 22, wherein the predetermined values include zeros.

24. (Canceled)

25. (Original) The method as in Claim 1, wherein providing the error indicator includes generating an interrupt.

26. (Original) The method as in Claim 1, wherein providing the error indicator includes setting a flag.

27. (Original) The method as in Claim 26, wherein separate flags are used for different error identifiers.

28. (Original) The method as in Claim 27, further including polling separate flags to determine the error.

29. (Original) The method as in Claim 28, wherein a driver is used to perform the polling.

30. (Original) The method as in Claim 1, wherein the transformed error correction data is related to multimedia data.

31. (Original) The method as in Claim 30, wherein the multimedia data includes video data.

32. (Original) The method as in Claim 31, wherein the transformed error correction data is discrete cosine transformed (DCT) data relating to video error correction data.

33. (Original) The method as in Claim 1, further including using a predetermined state when an error has been determined.

34. (Original) The method as in Claim 33, wherein the predetermined state includes providing a set of error correction data filled with predetermined values.

35. (Original) The method as in Claim 34, wherein the predetermined values include a set of error correction data filled with zeros.

36. (Original) The method as in Claim 35, wherein the error characteristic includes errors during the submission of processed transformed error correction data.

37. (Original) A method comprising:
performing error detection on received transformed data;
determining if an error has been found in the transformed data;
determining if the error is associated with a set of protected data; and
identifying a channel associated with the error if the error is associated with a set of protected data.

38. (Original) The method as in Claim 37, wherein determining an error includes identifying an error flag which has been set.

39. (Original) The method as in Claim 38, wherein a plurality of flags is polled to determine an error has occurred.

40. (Original) The method as in Claim 37, wherein determining an error includes receiving an interrupt indicating an error has occurred.

41. (Original) The method as in Claim 37, wherein determining if the error is associated with a set of protected data includes identifying an encryption key assigned to the set of protected data.

42. (Original) The method as in Claim 41, wherein identifying a channel associated with the error includes identifying a channel assigned an encryption key register.

43. (Original) The method as in Claim 37, further including performing corrective measures to reduce errors related to new data.

44. (Original) The method as in Claim 43, wherein corrective measures include clearing data buffers.

45. (Original) The method as in Claim 43, wherein corrective measures include re-authenticating encryption.

46. (Currently Amended) A computer readable medium tangibly embodying a program of instructions to manipulate a data processor to:

determine if an underflow error or an overflow error has occurred, wherein the error is related to transformed error correction data; and

apply corrective measures when an underflow error or an overflow error has occurred.

47. (Currently Amended) The computer readable medium as in Claim 46, wherein determining if the underflow error or overflow error has occurred includes detecting an interrupt generated in response to an underflow error or overflow error.

48. (Currently Amended) The computer readable medium as in Claim 46,wherein determining if the underflow error or overflow error has occurred includes determining if a flag has been set in response to an underflow error or overflow error.

49. (Original) The computer readable medium as in Claim 48, wherein the flag is cleared once it has been read.

50. (Original) The computer readable medium as in Claim 48, wherein individual flags of a plurality of flags are polled to determine if an error has occurred.

51. (Currently Amended) The computer readable medium as in Claim 50, wherein the individual flags relate to different error[[-]]_characteristics.

52. (Currently Amended) The computer readable medium as in Claim 51, wherein the error[[-]]_characteristics include error types.

53. (Canceled)

54. (Canceled)

55. (Currently Amended) The computer readable medium as in Claim 46, further including determining an error characteristic associated with the underflow error or overflow error.

56. (Original) The computer readable medium as in Claim 55, wherein corrective measures include clearing data buffers.

57. (Original) The computer readable medium as in Claim 46, further including identifying errors related to protected data.

58. (Original) The computer readable medium as in Claim 57, wherein corrective measures include initiating re-authentication.

59. (Currently Amended) A system comprising:

a data processor having an I/O buffer;

a memory having an I/O buffer coupled to the I/O buffer of the data processor, the memory capable of storing code to control said data processor to:

determine if an underflow error or overflow error has occurred, wherein the underflow error or overflow error is related to transformed error correction data; and

apply corrective measures when an underflow error or overflow error has occurred;

hardware coupled to said memory, said hardware including;

an inverse transform component to:

receive transformed error correction data, wherein the transformed error correction data is related to a set of image data; and

process said transformed error correction data to generate inverse transformed results;

determine if an error characteristic of the transformed error correction data has occurred; and

provide an error indicator when it is determined an error characteristic has occurred; and

a motion compensation processing component, wherein the motion compensation processing component to:

receive the motion compensation vector data, wherein the motion compensation vector data is related to said set of image data;

retrieve the inverse transformed results related to the set of image data, based upon the step of receiving motion compensation vector data; and

process the motion compensation vector data and the inverse transformed results to generate at least part of an image.

60. (Original) The system as in Claim 59, wherein the transformed error correction data includes a plurality of channels of transformed error correction data.

61. (Original) The system as in Claim 60, wherein the error characteristic includes identifying a transformed error correction data channel associated with errors.

62. (Currently Amended) The system as in Claim 59, wherein error[[-]]_characteristics include error types.

63. (Canceled)

64. (Canceled)

65. (Original) The system as in Claim 59, wherein the transformed error correction data includes DCT image data.

66. (Original) The system as in Claim 59, wherein the generated inverse transformed results represent a predetermined set of data when an error has occurred related to the transformed error correction data.

67. (Original) The system as in Claim 59, wherein providing the error indicator includes generating an interrupt.

68. (Original) The system as in Claim 59, wherein providing the error indicator includes setting a flag.

69. (Original) The system as in Claim 59, wherein corrective measures include clearing data buffers associated with transformed error correction data.